

# Problem with popen

---

*Source:* <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2007-10/msg00235.html>

---

- *From:* hg@xxxxxxxxxxxxxxxxxx
  - *Date:* Mon, 15 Oct 2007 07:39:18 -0700
- 

I'm having a problem with popen which is quite hard to explain. The behavior I'm seeing looks fairly random but must be related to a resource leak of some kind.

Under linux x86 I'm running a stress test which opens some tcp server applications and then about 70 clients in each their thread. The main thread is then spawning those new threads and waiting for them to join again. All this works perfectly fine. In this process the main thread additionally stops the external server application and restarts it at random intervals. This is to test how the clients behavior in a stressful situation when the server goes down behind their back. The clients connect using tcp as well and are just a library used from inside the application. One of the clients is a external application which is spawned as a synchronous process using popen, fread and pclose. The server however is started as an async process using fork and then execl.

What I'm seeing is that after 20 minutes run when everything is done and everything has settled then popen doesn't work anymore. When I start up another "tcp statistics client" using the same method of popen, fread and pclose then popen just opens the process, fread returns NULL and pclose returns without error. So obviously it may seem as the process would be mal functioning. However manual tests have shown that if the same external process is run on commandline then it works perfectly fine while fread still returns NULL.

So what I'm thinking is that some internal buffer got overflow and doesn't have capacity for more text. I wonder if this could be the result of a resource leak in my code or not.

Also what I notice is that when less than 70 threads are spawned then it's more likely to work. Finally when the servers are not restarted (async code not invoked) then it's also more likely to succeed.

It's a kind of weird problem but I was initially suspecting some stream not to be flushed. fflush didn't help anything though. Maybe it's related to not-closed handles with execl code.

At the end I've posted the C++ of this code for spawning and closing

## Problem with popen

sync and async processes.  
Any input is welcome.

Thanks.

— Henrik

```
-----  
  
CProcess::CProcess(string strCommand, enum SyncType eSyncType, int  
nTimeout)  
{  
    m_strCommand = strCommand;  
  
    m_bStarted = false;  
    m_eSyncType = eSyncType;  
}  
  
CProcess::~~CProcess()  
{  
    Stop();  
}  
  
bool CProcess::Run()  
{  
    if (m_bStarted)  
        return false;  
  
    int nNoWhere;  
  
    if (m_eSyncType == SYNC)  
    {  
        if ((m_fp = popen((char *) m_strCommand.c_str(), "r"))
```